What is claimed is:

1. A method of formal verification of a system design, wherein the system is defined by a set of automata, each having a set of states, the method comprising:
performing a first verification of a system, beginning with an initial state and finding successor states;
finding an under-defined state of the system;
saving execution traces leading up to the under-defined state;
further defining the under-defined state, thereby generating a newly specified state; and
performing a second verification of the system beginning with the newly specified state, using the saved execution traces, and finding successor states.

2. The method of claim 1, wherein performing a verification of the system comprises conducting a depth-first search through a state-space of a product automaton of the system.

3. The method of claim 1, wherein the set of automata comprises more than one automaton.

4. The method of claim 1, wherein further defining the under-defined state comprises specifying a control action for the under-defined state.

5. A method of formal verification of a system design, wherein the system is defined by a set of automata, the method comprising:
verifying a first instance of the system design;
saving verification data from the verification of the first instance of the system design, wherein the verification data comprise results of calculations used to verify the first instance of the system design;
modifying the system design, thereby generating a second instance of the system design; and
verifying the second instance of the system design using the saved verification data.

6. The method of claim 5, wherein verifying a first instance of the system design comprises checking a logical model corresponding to the system design.

7. The method of claim 6, wherein modifying the system design further comprises adding a design element to the logical model.

8. The method of claim 7, wherein saving verification data comprises saving results from the checking of the logical model.

9. The method of claim 5, wherein saving verification data comprises saving a set of paths that the system can follow in the first instance of the system design.

10. A method of formal verification of a system defined by a set of automata, the method comprising:

performing a first verification of the system, wherein the first verification comprises generating a partial solution pertaining to a first portion of the set of automata and generating a partial solution pertaining to a second portion of the set of automata;

modifying the system, wherein modifying the system comprises modifying one or more automata of the first portion of the set of automata without modifying any automaton of the second portion of the set of automata; and

performing a second verification of the system after modifying the system, wherein the second verification comprises generating a partial solution pertaining to the first portion of the set of automata and using the partial solution pertaining to the second portion of the set of automata generated from the first verification.

11. The method of claim 10, wherein performing the first verification further comprises generating a partial solution pertaining to a third portion of the set of automata.

12. The method of claim 11, wherein modifying the system further comprises modifying one or more automata of the third portion of the set of automata and wherein

performing the second verification further comprises generating a partial solution pertaining to the third portion of the set of automata.

13. The method of claim 11, wherein modifying the system further comprises modifying one or more automata of the first portion of the set of automata without modifying any automaton of the third portion of the set of automata and wherein performing the second verification further comprises using the partial solution pertaining to the third portion of the set of automata generated from the first verification.

14. The method of claim 10, wherein modifying an automaton comprises further specifying an under-defined state of that automaton.

15. A computer-usable medium having computer-readable instructions stored thereon adapted to cause a processor to perform a method, the method comprising:

performing a first verification of a system, beginning with an initial state and finding successor states;

finding an under-defined state of the system;

saving execution traces leading up to the under-defined state;

further defining the under-defined state, thereby generating a newly specified state; and

performing a second verification of the system beginning with the newly specified state, using the saved execution traces, and finding successor states.

16. The computer-usable medium of claim 15, wherein performing a verification of the system comprises conducting a depth-first search through a state-space of a product automaton of the system.

17. The computer-usable medium of claim 15, wherein the set of automata comprises more than one automaton.

18. The computer-usable medium of claim 15, wherein further defining the under-defined state comprises specifying a control action for the under-defined state.

19. A computer-usable medium having computer-readable instructions stored thereon adapted to cause a processor to perform a method, the method comprising:

verifying a first instance of the system design;

saving verification data from the verification of the first instance of the system design, wherein the verification data comprise results of calculations used to verify the first instance of the system design;

modifying the system design, thereby generating a second instance of the system design; and

verifying the second instance of the system design using the saved verification data.

20. A computer-usable medium having computer-readable instructions stored thereon adapted to cause a processor to perform a method, the method comprising:

performing a first verification of a system defined by a set of automata, wherein the first verification comprises generating a partial solution pertaining to a first portion of the set of automata and generating a partial solution pertaining to a second portion of the set of automata;

determining that one or more automata of the first portion of the set of automata have been modified without modifying any automaton of the second portion of the set of automata; and

performing a second verification of the system, wherein the second verification comprises generating a partial solution pertaining to the first portion of the set of automata and using the partial solution pertaining to the second portion of the set of automata generated from the first verification.